**MATLAB App 2.0 User Guide**

To summarize and visualize all the findings of this research, a culminating application is developed using the MATLAB App Designer. The following sections provide a user guide for the MATLAB application. To run the following apps, you will need to be in the App 2.0 directory in MATLAB.

main.mlapp

Chart

Description automatically generated

Figure 1. User interface of the main application

As shown in Figure 1, the interface of the main application allows you to visualize the efficiencies under 8 albedos using the optimal NN from hyperparameter tuning from this study – radbas and 38 hidden nodes, or to load the user’s own trained network by clicking on the Import NN button on the top left corner of the app. The Curr NN selected text field would show the name of the selected NN (default is radbas\_38\_4.7e-04.mat).

Using the default or loaded NN, the user can adjust the 6 sliders corresponding to the 6 geometric parameters (l\_FT, l\_PT, l\_BT, l\_FB, l\_HB, l\_PB) whose relative positions in the 4T APTSC is shown on the image at the bottom left corner. For consistency, the color code of each layer in the image matches the color of the text and sliders. The text field above the sliders give a numerical value of that geometric parameter in nm.

In the top right corner of the app, a plot of efficiency vs. albedo visualizes the output of the NN given the 6 geometric inputs specified by the sliders. As the user moves the slider, this plot would change accordingly.

The bottom right corner of the app is a gauge that illustrates the percentage efficiency of the albedo specified in the dropdown text box to its left. The gauge is able to show the efficiency under each of the 8 albedos with the numerical value of the efficiency placed at its lower center.

The main app also directs the user to train their own NN through the button at the top left corner. Once the user clicks this button, a separate dialogue app would show up for the user.

train.mlapp

Graphical user interface, application

Description automatically generated

Figure 2. User interface of the train application

Figure 2 shows the user interface of the train application where the user can train their own NN by loading their own training and test datasets. This window contains stepwise instructions from loading data to selecting hyperparameters to training and saving the NN.

Step 1. Data loading

There are two ways to load the training and test data. First, you can fast load the data by clicking on the green *Fast Load Data Folder* button. It will direct you to a dialogue for you to choose the folder containing all the input and output training and test datasets. The text files in the folder needs to satisfy the following format for the app to work properly;

1. There should be 6 input text files named l\_BT.txt, l\_FB.txt, l\_FT.txt, l\_HB.txt, l\_PB.txt, l\_PT.txt. Each text file contains 1 column of geometric property of the PSC in nm.
2. There should be 8 output text files named Output\_02.txt, Output\_con.txt, Output\_DG.txt, Output\_GG.txt, Output\_RB.txt, Output\_RS.txt, Output\_S.txt, and Output\_WS.txt corresponding to the 8 albedos. Each text file contains 3 columns of comma separated data: the first column is the top cell efficiency, second column is bottom cell efficiency, and the third column is overall efficiency.
3. The test data should be 1 text file named test\_IO.txt. There should be 14 columns of comma separated data. The first 6 columns are input data of the geometric properties (l\_FT, l\_PT, l\_BT, l\_FB, l\_HB, l\_PB in that order) and the last 8 columns labelled efficiency data of the 8 albedo conditions (in the order of: 0.2 albedo, concrete, dry grass, green grass, red brick, roof shingle, snow, and white sand)

The second way to load the data is by loading each input and output text files separately through the dropdown menus on the top right corner of the app. This method is slower, but it allows you to load your data if they are not in the same folder.

Step 2. Select hyperparameters

This step allows the user to select 3 hyperparameters to train the network: number of hidden layers, the size of each layer, and the activation function of each layer. Based on the selected number of hidden layers, the app shows a corresponding number of layer size and activation function selections.

Graphical user interface, application

Description automatically generatedStep 3. Train and save your NN

After you have loaded the data and selected all your hyperparameters, click the *Start Training* button to train your NN. While the app is working on it, you will see a yellow progress light with the sign: In Progress. After your NN finishes training, the progress light will turn green and the sign would say “Done”. There would also be a reminder that the trained NN is saved to the current directory.

Figure 3. NN training done and saved to current directory

Step 4. Reset and Return to Main

Graphical user interface, text, application

Description automatically generatedTo reset the hyperparameter selection and retrain your NN, simply click on the *Reset* button, and the app will reset all hyperparameter values and wait for you to enter new values, as shown in Figure 4, If you wish to quit this app or return to the main app, click the *Return to Main* button at the lower right corner.

Figure 4. Resetting the hyperparameters to train another NN

genetic.mlapp

This section of the app allows the user to apply and visualize the genetic algorithm to optimize efficiency and obtain the corresponding geometric parameters.

Step 1. Import NN

The user can load a trained NN of their own or use the default preloaded 1 HL NN with radbas and 38 hidden nodes.

Step 2. Select the genetic algorithm hyperparameters

The use can select the hyperparameters of the GA including crossover fraction, elite count, and population size. Please make sure that your elite count does not exceed the population size because the elites come from the population. In addition, the user also specifies the stopping conditions including the max stall generations and the max training time for the GA.

Step 3. Train and visualize the optimal results

Graphical user interface

Description automatically generatedBy clicking the *Start Training* button, the GA starts, and the progress button will be yellow with the sign *In Progress*, as Figure 5 shows. After the GA completes optimization, the progress button becomes green, and the sign will read *Done.* The two plots on the right of the app will visualize the optimization results, as illustrated by Figure 6. The top graph plots the optimal efficiencies for all 8 albedos. The bottom plot illustrates the thicknesses of the 6 optimal geometric parameters under the albedo specified by the dropdown menu. By hovering over the specific points, the user is able to read the corresponding coordinates of that point.

Figure 5. Genetic algorithm app during training

Step 4. Visualize the parallel coordinate plot

To visualize the GA optimization process, the user will just have to click on the *Parallel Plots* button at the bottom of the app. This will create a separate dialogue displaying the parallel coordinate plot under the albedo specified by the dropdown menu. The parallel plot will highlight the high, medium, low, optimal, and base efficiencies.

Graphical user interface

Description automatically generated

Figure 6. Genetic algorithm app after training finishes